

REMARKS

This is in full and timely response to the above-identified Office Action. The above listing of the claims supersedes any previous listing. Favorable reexamination and reconsideration are respectfully requested in view of the preceding amendments and the following remarks.

Claim 1-23 remain pending in this application.

Firstly, the rejections would indicate that the claim amendments entered via the response filed on May 22, 2007, have not been acknowledged. The fact that Fisher et al. qualifies as 102(b) art as different from 102(e) is noted, however the comments at the top of page 3 of this office action indicated that the claim amendments have not been taken into consideration. The amendments are also seen to negate the rejections under § 101 and §112. In connection with the § 112 rejection attention is called to claim 7 which has been amended to obviate the lack of clarity introduced by the "said models."

As to the rejection of claims 1-9, 11-19 and 21-23 under 35 USC § 103(a) as being unpatentable over Fisher et al. in view of Larson, the applicants firstly take the position that Fisher et al. does not explicitly disclose the claimed "generating a mapping between said source model and said target model." At best there is a presumption that the mapping must "be[sic] existed" in order to the processor to correctly copied to the second data structure. This, of course, is an attempt to use "inherency."

As the Examiner is required to know, for inherency to hold it must occur in each and every instance, not just in a few cases or even most cases. This requirement is such as to render the position that "mapping must be existed" (presumably – must have existed) untenable unless it can be shown that this "mapping", which is not even mentioned in the reference, would occur each and every time data is copied in the manner indicated in Fisher et al.

A clear and intelligible showing that the above-mentioned situation would exist in each and every instance must be made or the rejection must be withdrawn.

The applicant has made every effort to understand what the Examiner is attempting to express, however, it is not incumbent on the applicant or his/her representatives, to second guess the rejection and attempt to create sense of inappropriately expressed rejections/statement upon which examination of the claimed subject matter hinges.

Indeed, the language of the reference which is relied upon for the sake of rejection is unclear. For example, the statement "a first data structure is declared that a compiler, for example a C compiler, interprets as packed" is not clear. Certainly, a declaration that first data is packed does not amount to anything near disclosure of "generating a source model of a source format element." At best what is disclosed is that a declaration of data being packed, is made. How this generates a source model is not at all clear and is not, in the applicants' opinion, suggestive of the claimed subject matter.

Note must be had to the fact that this rejection is made under the § 103 statute and as the Examiner is aware, in order to establish a *prima facie* case of obviousness, it is necessary to show that the hypothetical person of ordinary skill would, without any knowledge of the claimed subject matter and without any inventive activity, be motivated to arrive at the claimed subject matter given the guidance of the cited references when each is fully considered as statutorily required.

The Examiner is requested to explain how a declaration that a first data structure is packed – discloses the claimed "generating a source model of a source format element." The Examiner is also requested to acknowledge that the rejection cannot be a quasi § 102 rejection wherein highly obtuse interpretations of disclosure can be made sans the common sense that must accompany the consideration by the hypothetical person of ordinary skill.

A similar request is made with respect to the assertion that the claimed "generating a target model of a target format element" is in fact disclosed/suggested by "a second data structure is declared that the compiler interprets as unpacked." Applicants do not see how declaring a packed status discloses/suggests "generating a target model . . ." It is submitted a clear and concise explanation is required before the above assertions can be validated.

In connection with the citation of Larson, it is noted that only two lines on one column of this reference are relied upon and that, without any showing to the contrary, the remainder of the Larson disclosure is flatly ignored. The disclosure of a code generator generating a translator program is lifted out of the Larson reference and used in a situation wherein code generation is not mentioned

More specifically paragraph [0027] of Fisher et al. discloses:

[0027] FIG. 7 is a flowchart of an unpacking method in accordance with one aspect of the invention. The method of FIG. 7 may be used, for example, to unpack one or more data structures contained within a packed ACPI table. At 705, a first data structure is declared that a compiler, for example a C compiler, interprets as packed (refer to the earlier description associated with FIG. 2A and FIG. 2B for an explanation of packed versus unpacked data structures). At 710, a second data structure is declared that the compiler interprets as unpacked. At 715, the data type associated with the first data structure declared at 705 is applied to a pointer referencing the original packed data structure. That is, the compiler is directed to treat the data referenced by the pointer as if it were of the data type associated with the first data structure. In the C programming language, this may be accomplished **by means of a cast, in which a data type is applied to another data structure having a different innate data type.** For example, the C-language cast "(mystruct *)" directs the compiler to treat the object following **the cast as a pointer to an object of data type "mystruct."** At 720, the data is copied from the original packed data structure to the second, unpacked data structure **using the pointer receiving the cast** at 715. The pointer receiving the cast at 715 causes the compiler to treat data read from the original packed data structure at 720 as packed data so that it may be correctly copied to the second, unpacked data structure. Once the desired data has been copied, the method returns control to the calling program at 725. (Emphasis added)

Thus, it is not seen that it would be obvious to modify Fisher et al.'s approach to generate a translator program in light of the disclosure of Larson. To the contrary, if a translator program was required, Fisher et al. would be dropped and the system disclosed Fisher et al. would be adopted.

Conclusion

To the extent that the Examiner is apparently examining the wrong set of claims, should he still hold that the cited art renders the claimed subject matter obvious in accordance with the § 103 statute wherein the each of the references must be considered for all that is disclosed therein, a further non-final office action is requested so that the issues which stand before the PTO can be clarified in writing and thus make complete the written record in clear and unambiguous terms.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 07-1337 and please credit any excess fees to such deposit account.

Respectfully submitted,
Dibyapran SANYAL *et al.*



Kenneth M. Berner
Registration No. 37,093

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, CO 80527-2400
703-684-1111 Telephone
970-898-0640 Telecopier
Date: October 10, 2007
KMB/KT/mkl/jlb